**To Supply Leftover Food to Poor**

**(Documentation)**

# Project Overview

The "To Supply Leftover Food to Poor" project is a Salesforce-based CRM solution designed to bridge the gap between food donors (such as restaurants, event organizers, and households) and the underprivileged who are in need of food. This CRM platform efficiently manages food donation requests, tracks available surplus food, schedules pickups, and coordinates deliveries to nearby shelters or distribution centers. The system ensures that excess food is utilized instead of wasted, promoting sustainability while addressing hunger in local communities.

**Key Features:**

* Donor registration and leftover food entry.
* Real-time tracking of food availability and expiry timelines.
* Automated matching of donors with nearby NGOs or shelters.
* Pickup and drop-off scheduling and management.
* Volunteer and delivery coordination.
* Reports and dashboards for monitoring donations, deliveries, and impact.

**Business Needs Addressed:**

* Reducing food wastage in urban areas.
* Providing a structured and transparent method to supply leftover food to those in need.
* Enhancing coordination between donors, NGOs, and delivery personnel.
* Enabling real-time updates and efficient resource management through a centralized system.

# Objectives

The primary objective of building the **"To Supply Leftover Food to Poor"** CRM system is to create a centralized, automated platform that connects food donors with NGOs, volunteers, and needy communities in real time. This CRM aims to streamline the process of collecting, managing, and distributing leftover food, ensuring timely delivery before spoilage and maximizing the impact of each donation. By digitizing the donation workflow, the system reduces manual coordination, improves transparency, and enhances accountability. From a business value perspective, the CRM promotes better stakeholder management (donors, volunteers, NGOs), enables data-driven decision-making through reports and dashboards, and supports scalability of food distribution operations in multiple regions—ultimately creating a sustainable ecosystem to reduce hunger and food waste simultaneously.

# Phase 1: Requirement Analysis & Planning

## Understanding Business Requirements

The core business need is to efficiently collect and distribute leftover food to underprivileged individuals, while minimizing food wastage. Various stakeholders—including food donors (restaurants, caterers, individuals), NGOs, and volunteers—face challenges such as lack of coordination, delays in pickups, and food spoilage. This CRM aims to solve these problems by providing a centralized, real-time system to manage donor entries, schedule pickups, assign delivery volunteers, and track donations.

**User Needs:**

* Donors need an easy way to register and log available food with location and expiry time.
* NGOs and shelters need to receive food before it spoils and manage their storage capacity.
* Volunteers need clear pickup and delivery assignments with route information.
* Admins require real-time dashboards, reports, and alerts to manage operations efficiently.

## Defining Project Scope and Objectives

**Project Scope Includes:**

* User registration and role-based access (Donor, NGO, Volunteer, Admin).
* Form to log leftover food details (quantity, type, location, expiry time).
* Automated matching of food donations to nearby NGOs based on location and need.
* Scheduling and assigning delivery pickups to available volunteers.
* Notifications to all parties involved (confirmation, pickup updates, delivery status).
* Dashboard for tracking daily food donations, number of people served, and spoilage prevented.
* Report generation for impact analysis.

**Project Will Not Include:**

* Real-time GPS tracking.
* Integration with third-party food delivery apps (like Swiggy, Zomato).
* Payment or monetary donation features.

## Design: Data Model and Security Model

**Data Model (Key Objects):**

* **Donor\_\_c**: Stores donor details (Name, Contact Info, Address).
* **Food\_Donation\_\_c**: Stores each donation record (Food Type, Quantity, Expiry Time, Donor Lookup).
* **NGO\_\_c**: Stores NGO information and capacity.
* **Volunteer\_\_c**: Stores volunteer profiles and availability.
* **Delivery\_\_c**: Stores pickup and delivery schedules (linked to Food\_Donation\_\_c and Volunteer\_\_c).
* **Feedback\_\_c**: Collects feedback from NGOs after delivery.

**Security Model:**

* **Profiles and Roles:** 
  1. Donor: Can create/view their own donations.

○ NGO: Can view available food and confirm receipt.

○ Volunteer: Can view assigned deliveries.

○ Admin: Full access to all objects and records.

* **Sharing Rules:** 
  1. Org-wide defaults (OWD): Private for Food\_Donation\_\_c and Delivery\_\_c.

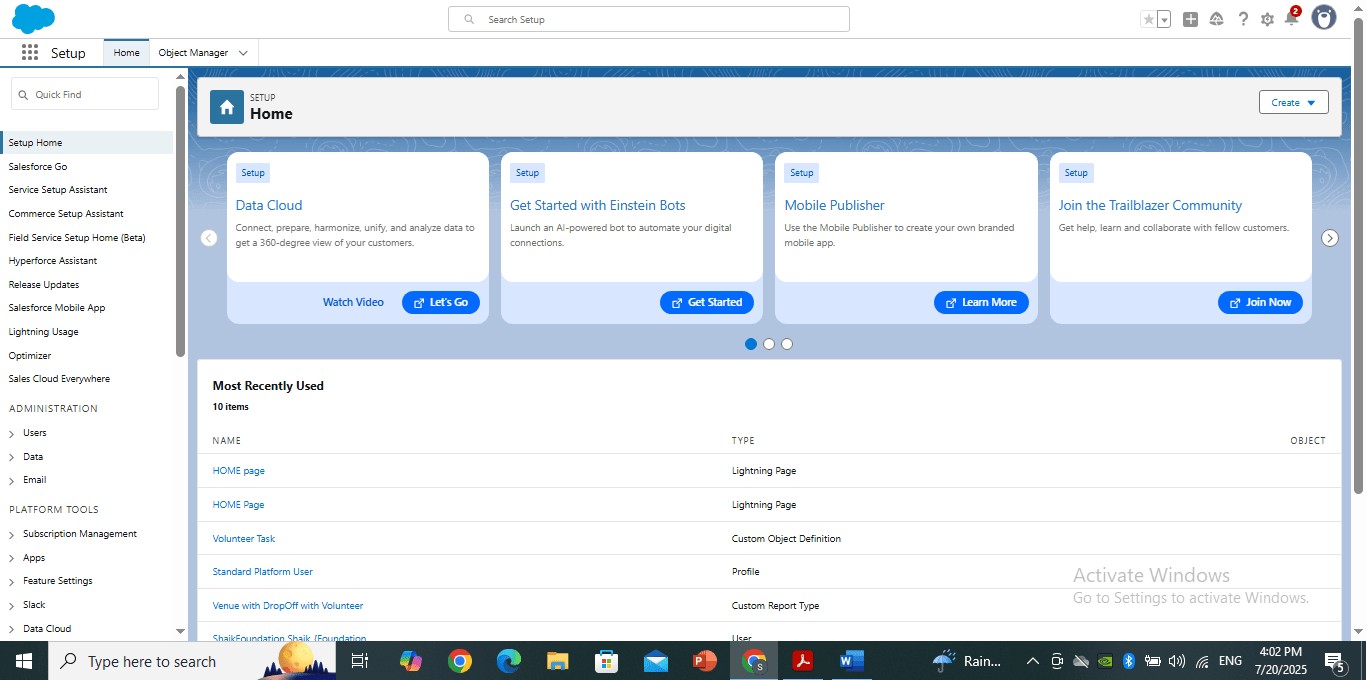
○ Sharing rules to allow NGO and Volunteer access to relevant records.

* **Field-level Security:** 
  1. Sensitive fields like personal contact details and delivery addresses are restricted from users who don’t need them.

# Phase 2:Salesforce Development - Backend & Configurations

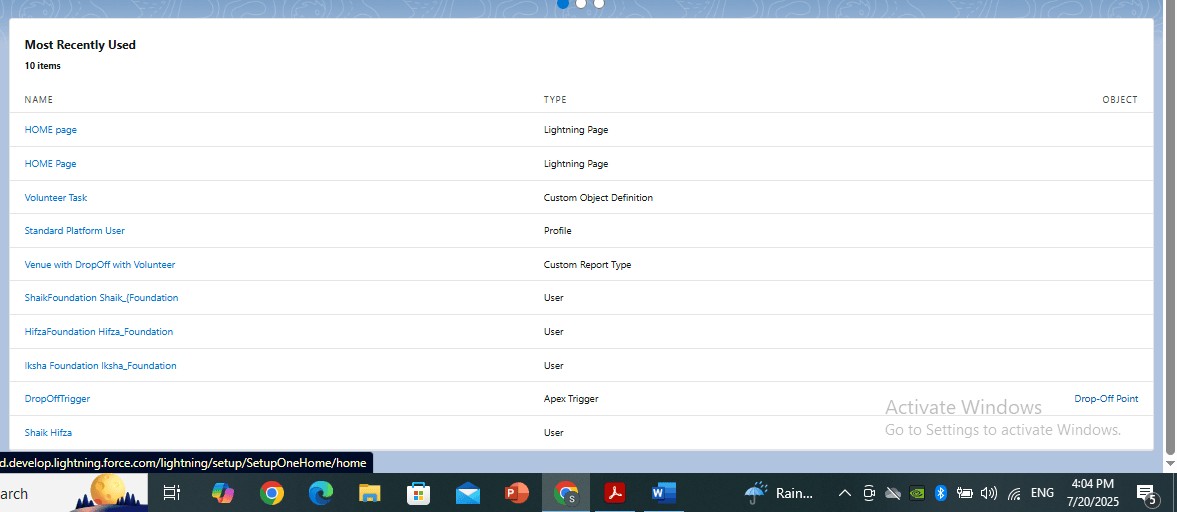
## Setup Environment & DevOps Workflow

Setting up a Salesforce Developer Org or Sandbox to develop and test the CRM before deploying it to production. The DevOps workflow involves tracking changes (metadata), using Change Sets, or tools like VS Code with Salesforce CLI to move components between environments safely and efficiently.



## Custom Objects & Fields

Custom Objects are database tables created to store specific data related to your business processes (like Donor, Food\_Donation, NGO). Custom Fields are added to store information such as quantity, food type, expiry date, etc., tailored to your app's needs.



## Validation Rules

Validation Rules ensure that users enter correct and meaningful data in records. They use formulas to check field values and show error messages if the data doesn't meet certain conditions.

Example: Preventing users from entering an expiry date in the past.

## Workflow Rules

Workflow Rules are used to automate simple tasks like sending email alerts, updating fields, or creating tasks when a record meets certain criteria.

Example: Send an email alert when a food donation is submitted.

## Process Builder

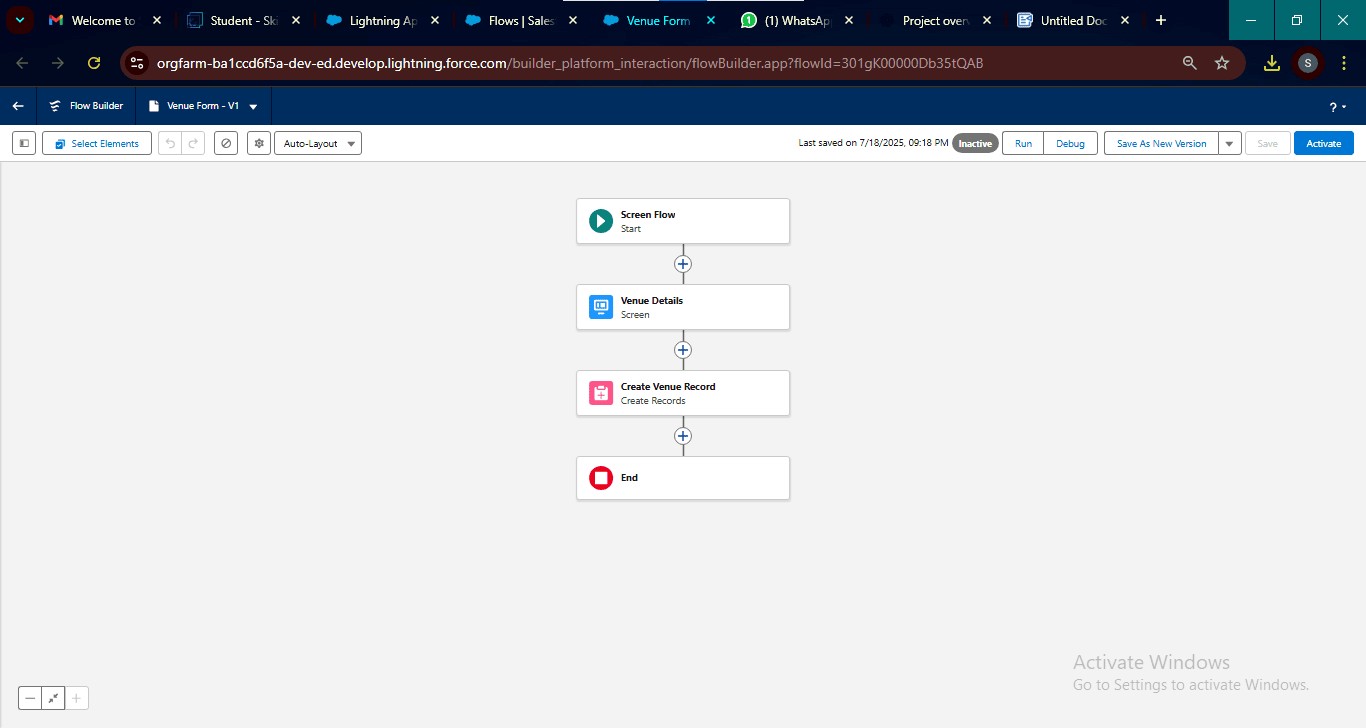
A visual tool that allows admins to automate multi-step business processes with conditions, actions, and updates without writing code.

Example: If a donation is confirmed and within 5 km of an NGO, assign it automatically.

## Flows (Record-Triggered Flow)

Flows are powerful automation tools that allow you to create logic for complex, multi-step actions. Record-Triggered Flows start automatically when a record is created or updated.

Example: Auto-create delivery and assign a volunteer when a donation is confirmed.



## Approval Process

A structured way to submit records for review and approval by one or more users. Used when a decision must be made before a record can move forward.

Example: Large food donations require approval by NGO admin before scheduling pickup.

## Apex Classes

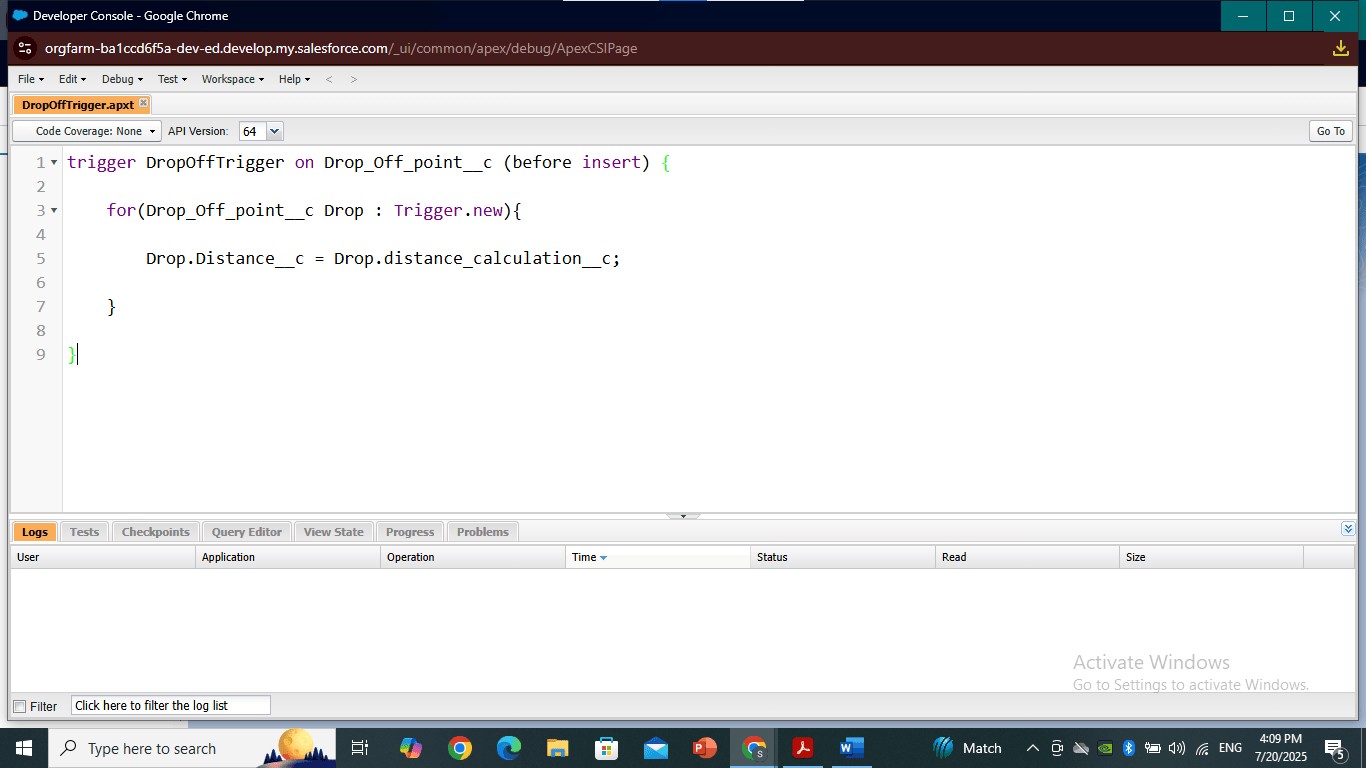
Apex Classes are code files written in Salesforce’s programming language (Apex) that contain reusable functions or logic to control complex business processes or integrations.

Example: A class that finds the nearest NGO using donor location and availability.

## Apex Triggers

Triggers are pieces of Apex code that run automatically when records are created, updated, or deleted. They allow developers to respond to data changes programmatically.

Example: Automatically creating a Delivery record when a Food Donation is confirmed.



## Asynchronous Apex

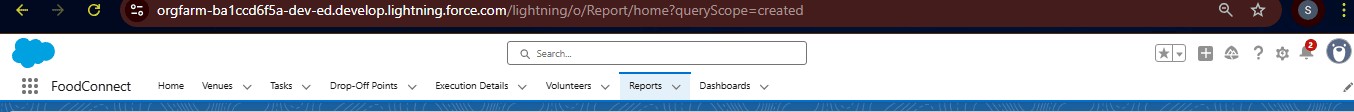
Asynchronous Apex (like @future, Queueable, Batch Apex) is used to perform long-running or background tasks without delaying user interactions.

Example: Sending bulk SMS/email to multiple NGOs about upcoming food donations.

# Phase 3: UI/UX Development & Customization

## 1. Lightning App Setup (via App Manager)

Creating a custom Lightning App through App Manager allows you to define a tailored workspace with navigation items (tabs) such as Donors, Food Donations, Volunteers, etc.



## 2. Page Layouts & Dynamic Forms

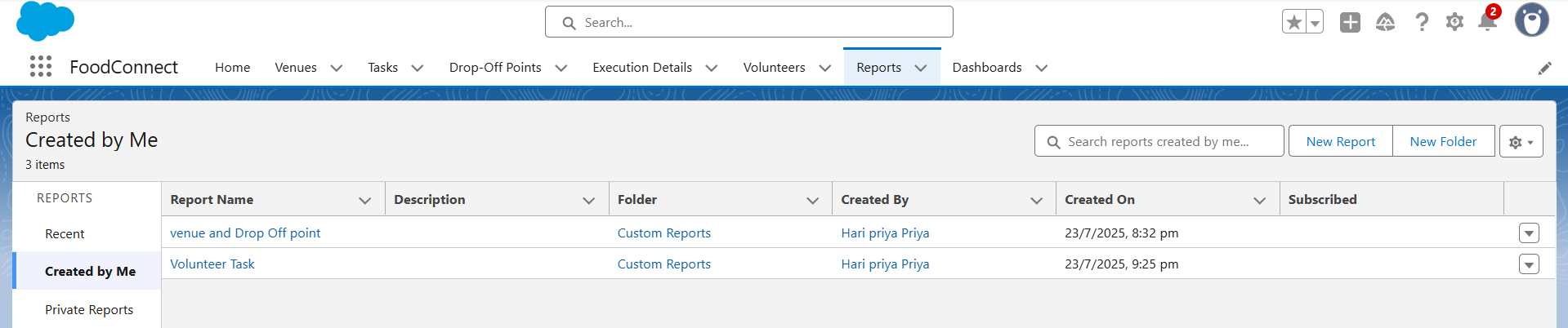
* **Page Layouts** determine how fields, buttons, and sections are arranged on record detail pages.
* **Dynamic Forms** allow showing/hiding fields based on certain conditions without needing multiple layouts.

## 3. User Management

Managing users includes creating user accounts, assigning Profiles, Roles, and Permission Sets to control data access.

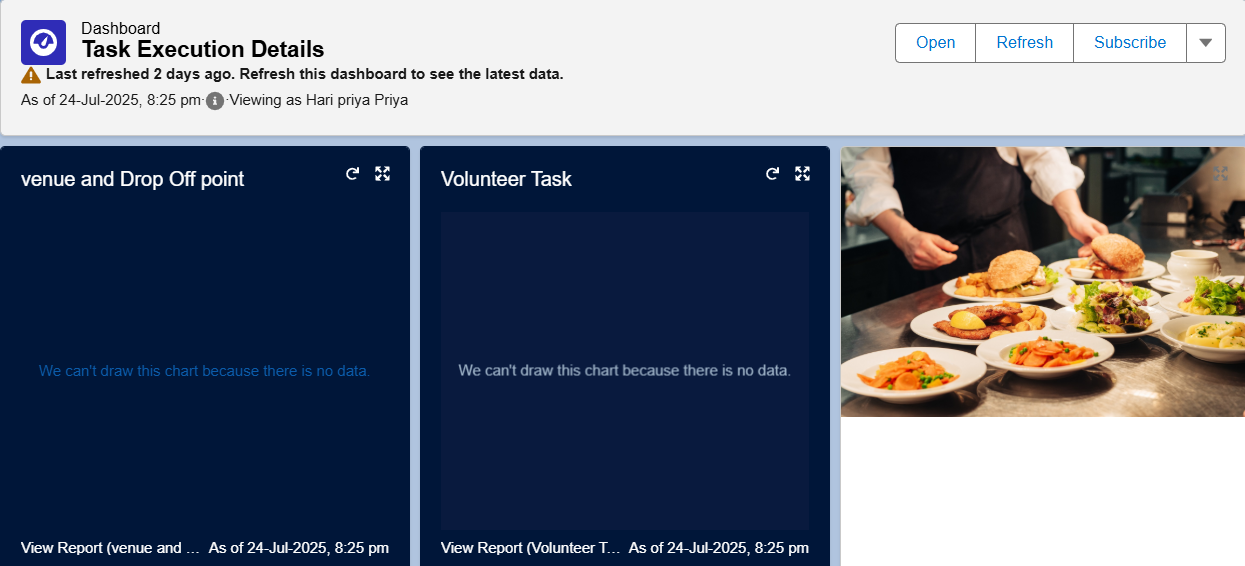
## 4. Reports and Dashboards

Reports help visualize data like the number of donations per day, top donors, or NGOs served. Dashboards summarize this data in charts, graphs.



## 5. Lightning Web Components (LWC) and Lighting Web pages

LWC is a modern Salesforce framework for building fast, reusable UI components. You may build custom components if standard UI doesn’t meet the need.



# Phase 4: Data Migration, Testing & Security

## 1. Data Migration Process

Data migration involves importing initial data like donor details, food entries, NGO contacts, and volunteer records into Salesforce.

**Tools Used:**

* **Data Import Wizard**: Used for simple imports (Donors, NGOs).
* **Data Loader**: Used for bulk import of large datasets (Food\_Donation\_\_c, Delivery\_\_c records).

## 2. Field History Tracking, Duplicate Rules, Matching Rules

**Field History Tracking**

Tracks changes made to selected fields, such as Quantity, Status, or Assigned\_Volunteer

**Duplicate Rules**

Prevents users from creating duplicate records (e.g., same donor with same contact).

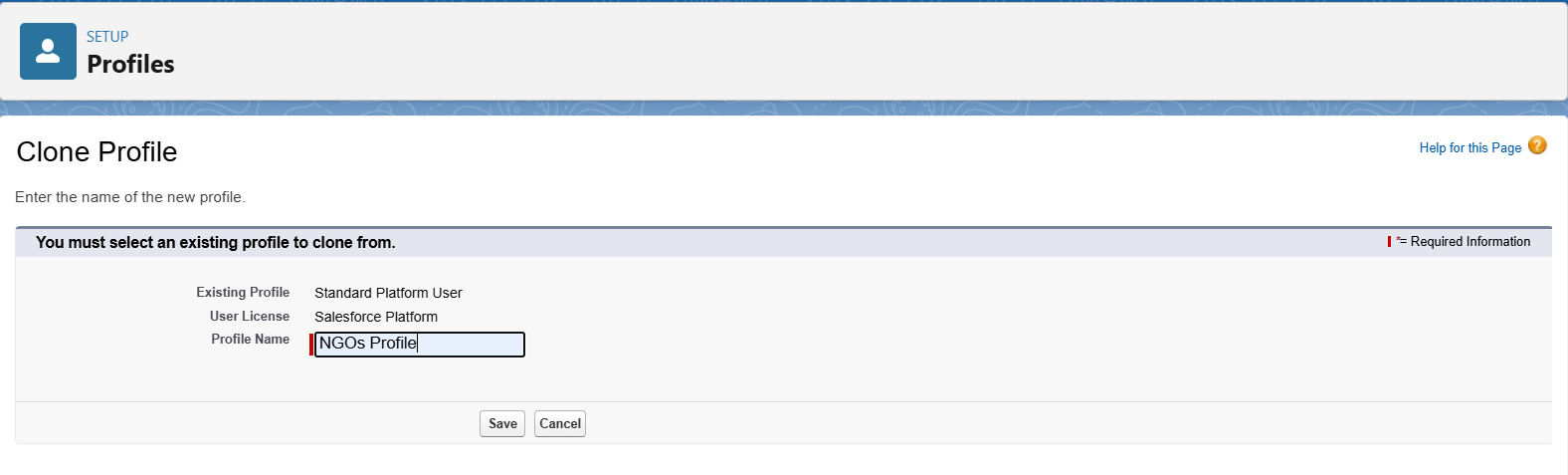
**Matching Rules**

Defines how Salesforce detects duplicates (based on email, phone, or custom logic).

## 3. Profiles, Roles, Role Hierarchy, Permission Sets, Sharing Rules

**Profiles**

Controls what users can do (object-level and field-level access). Example: Donor profile can only create/view their records.



**Roles & Role Hierarchy**

Defines data visibility and record-level access. Hierarchy allows senior roles to see data of users below them.

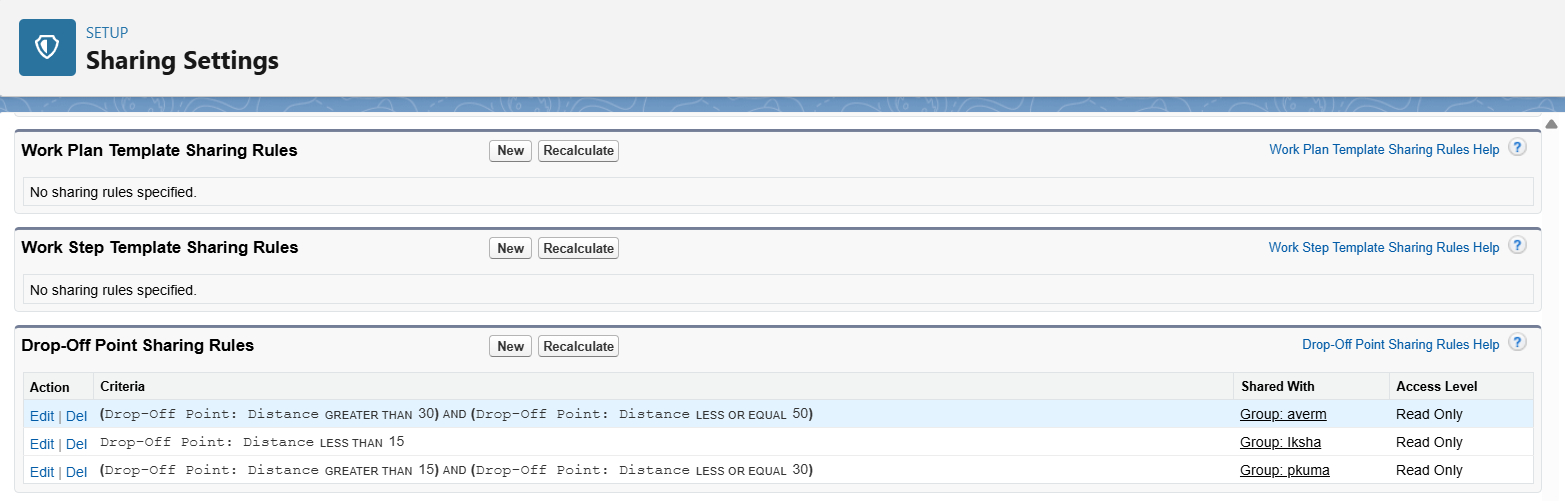
**Permission Sets**

Used to grant additional permissions to users without changing their profile.

**Sharing Rules**

Grants record access based on criteria or role.

Example: Volunteers can see Food\_Donation\_\_c records assigned to them.



## 4. Creation of Test Classes

Test classes are written in Apex to test code coverage of triggers, classes, and logic. Salesforce requires 75% minimum code coverage for deployment.

@isTest public class TestCreateDeliveryTrigger {

@isTest

static void testDeliveryCreation() {

Food\_Donation\_\_c fd = new Food\_Donation\_\_c(Name='Test Food',

Status\_\_c='Confirmed');

insert fd;

List<Delivery\_\_c> deliveries = [SELECT Id FROM Delivery\_\_c WHERE

Food\_Donation\_\_c = :fd.Id];

System.assertEquals(1, deliveries.size());

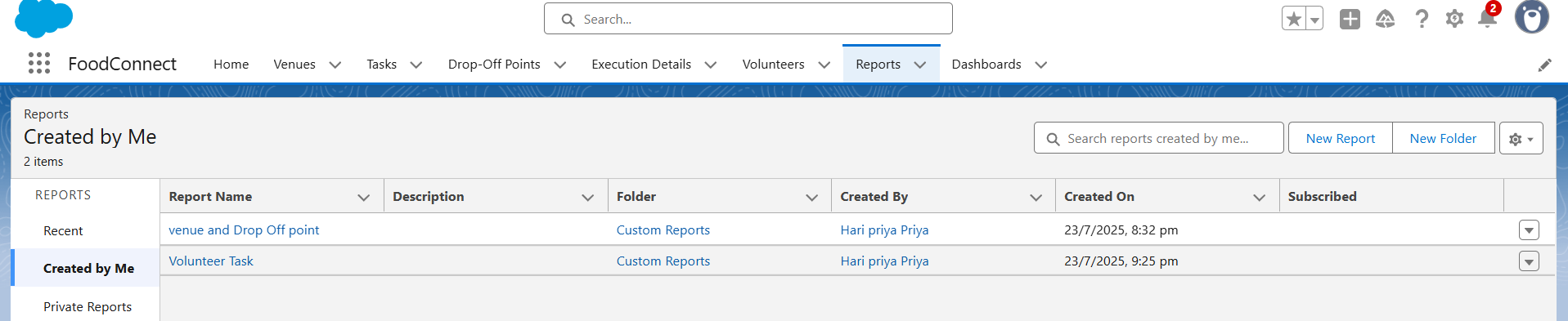
}

## 5. Test Cases for Salesforce Features

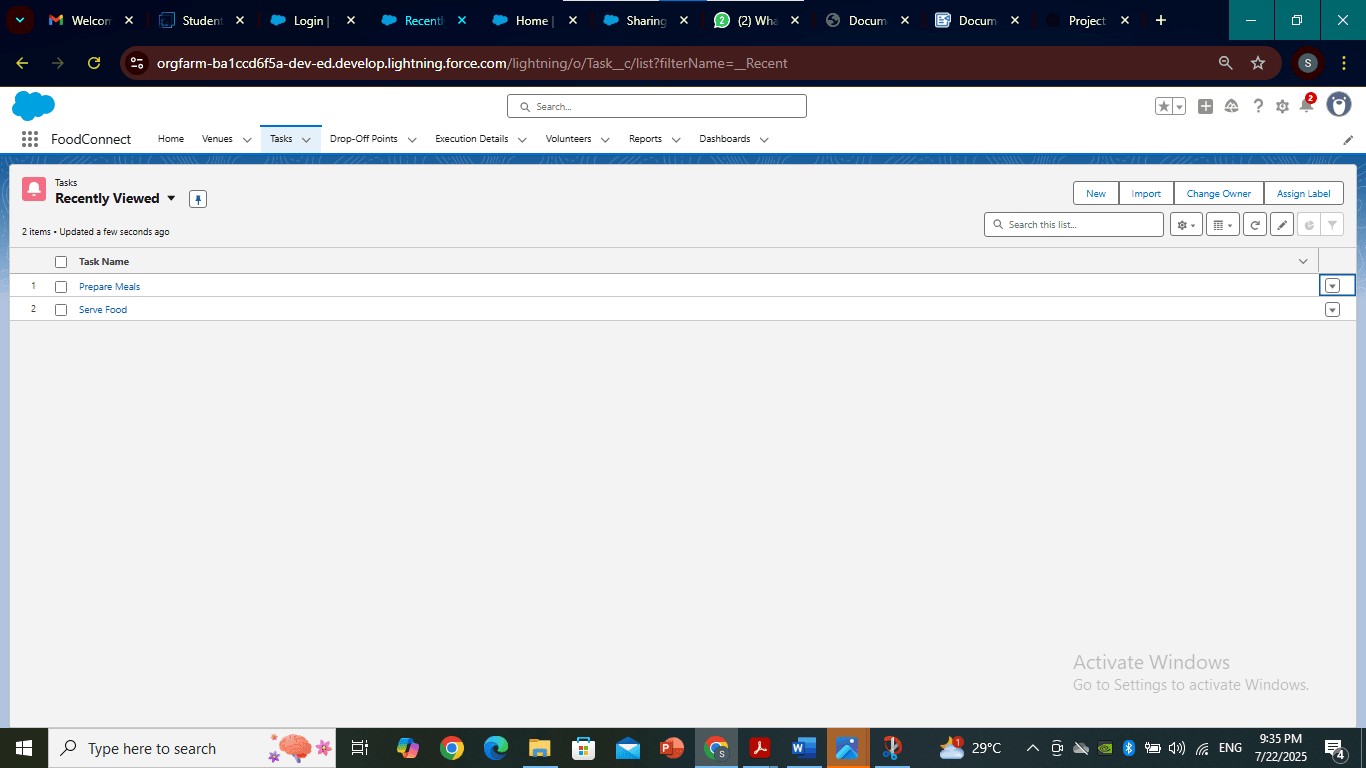
Volunteer,Organization and Execution Test Cases **Inputs:**



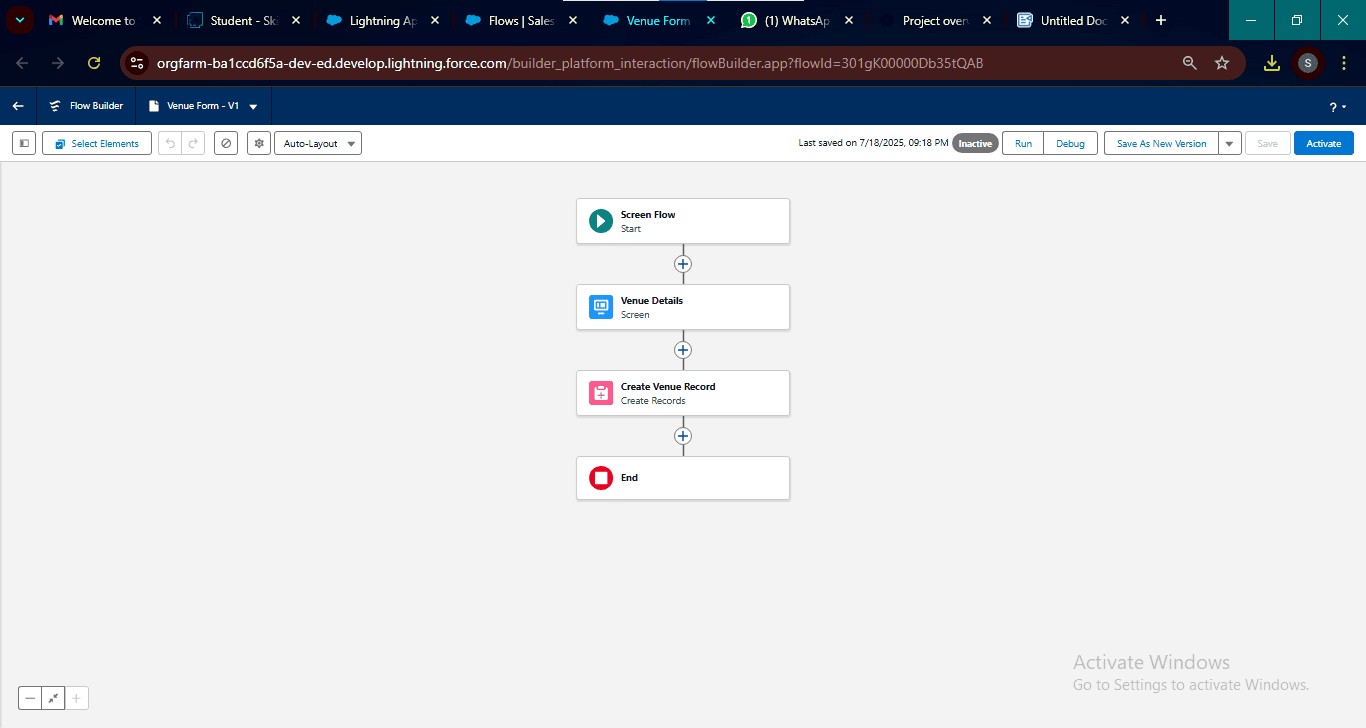
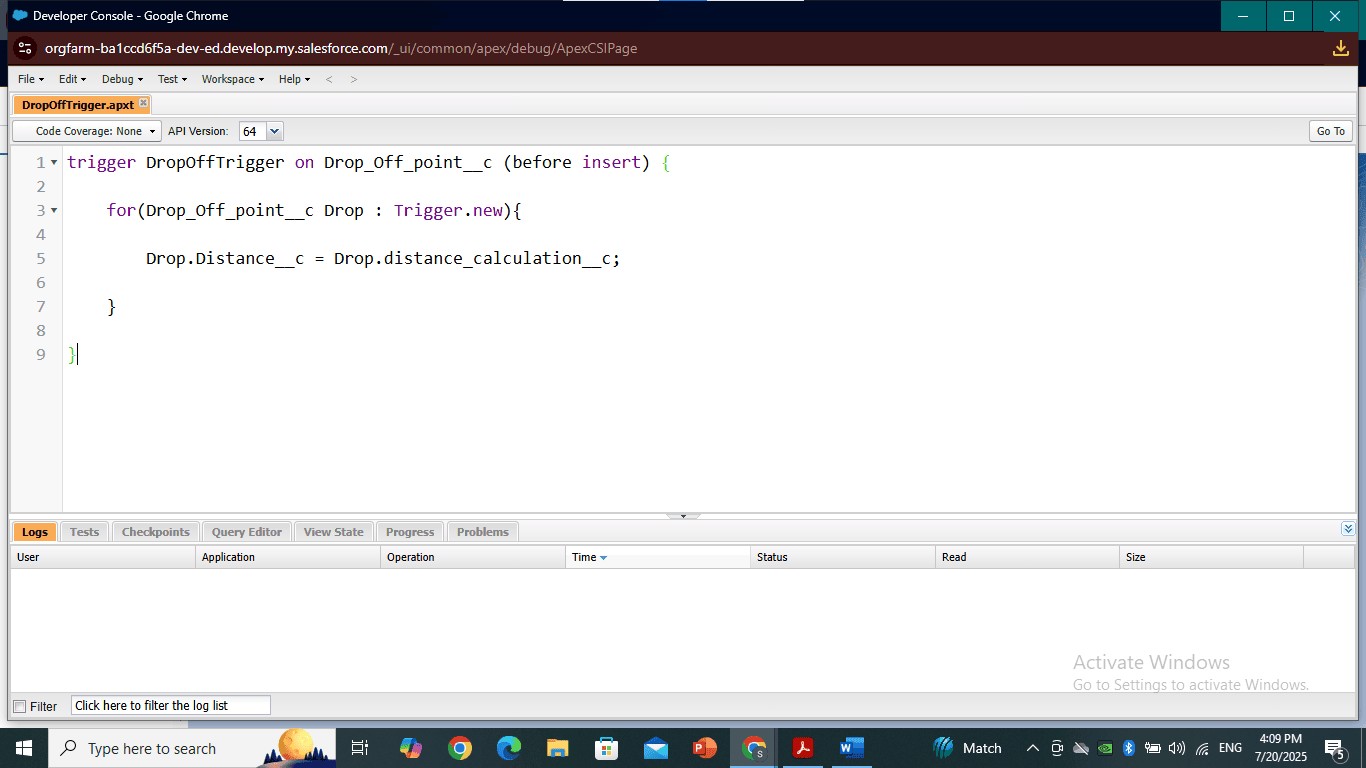
## Output TestCases ScreenShot



**Automatic Task Creation:**



**Flows and Triggers ScreenShots:**



# Phase 5: Deployment, Documentation & Maintenance

## 1. Deployment Strategy

The project was deployed using Change Sets, which allowed safe and structured migration of components from Sandbox to Production. All custom objects, automation, and Apex code were validated before deployment.

## 2. System Maintenance & Monitoring

The system will be maintained through regular checks using tools like Health Check, Scheduled Jobs, and Debug Logs. Admins will monitor user activity, system errors, and flow executions to ensure smooth operation.

## 3. Troubleshooting Approach

A step-by-step approach is documented to solve common issues using Debug Logs for errors, audit trails for tracking data changes, and a knowledge base for known problems like duplicate records or permission errors.

# Conclusion

The "To Supply Leftover Food to Poor" Salesforce project successfully provides a structured and automated platform to connect food donors with NGOs and volunteers, ensuring that surplus food is utilized effectively rather than wasted. Through custom objects, automation, and userfriendly interfaces, the system simplifies donation tracking, pickup scheduling, and delivery coordination. This CRM not only streamlines operations but also supports a greater social cause—reducing hunger and promoting sustainability.